

PDF SUB-PROCESO - COMPARADOR

Flow: Escalado Formatos · 5 similares + 1 arriesgado (factory_v4.html)

Station ID	comparador
Skill canónica	comparador-finalistas
Agent .md	.claude/agents/escalado_formatos/comparador-finalistas.md
Verifier L1	verify_6_finalists.py (5 similares + 1 arriesgado, todos awareness-match)
Supervisor L2 skill	Detectar si el comparador está siendo demasiado laxo o demasiado estricto
Modelo default	claude-opus-4-7 (msg 4658)
Master PDF flow	_documentacion/PDF_MAESTRO_ESCALADO_VIDEO_v16.pdf · Flow D "Concepto ganador ·
Generado	2026-05-09 20:26

Especificación canónica de la skill (verbatim del agente .md)

Esto es lo que el agente DEBE hacer literalmente. El supervisor de control de calidad valida cada output contra esta spec.

Agente COMPARADOR-FINALISTAS

Misión

De todos los candidatos que los 4 buscadores trajeron, elegir los 6 mejores: 5 que reproducen el patrón ganador (similares) + 1 que arriesga con un giro (variante para descubrir nueva ramificación escalable).

Inputs

- analysis_path: _analysis.json del video ganador.
- candidates_paths: lista de 4 paths _candidates_<buscador_id>.json.

Output

JSON _finalists.json:

```
{
  "timestamp": "2026-05-09T17:00:00Z",
  "product": "...",
  "ganador_ref": "<video_url ganador>",
  "finalistas": [
    {
      "rank": 1,
      "tipo": "similar",
      "video_url": "...",
      "video_path": "_buscar/_downloaded/abc.mp4",
      "match_score": 0.92,
      "match_breakdown": {
        "awareness": 1.0, "sof": 1.0, "jergas": 0.85, "deseo": 1.0,
        "avatar": 0.90, "temporalidad": 1.0, "fuerza_cambio": 0.88,
        "formato": 0.95
      },
      "por_que_similar": "Mismo Problem Aware + mismo deseo dolor cervical + mismo avatar 50+ + mismo f
      "speech_summary": "..."
    }
  ]
}
```

```

    },
    /* ... 4 más similares ... */
    {
      "rank": 6,
      "tipo": "arriesgado",
      "video_url": "...",
      "match_score": 0.74,
      "diferenciador": "Mismo deseo + mismo avatar PERO formato comparativa lateral (vs prótesis cirug)",
      "speech_summary": "..."
    }
  ],
  "rejected_count": 14,
  "rejected_reasons_summary": {"awareness_mismatch": 6, "avatar_outside": 4, "ugc_basico_saturated": 4}
}

```

Procedimiento

1. Cargar análisis ganador y consolidar candidatos de los 4 buscadores (deduplicar por video_id).

2. Para CADA candidato, calcular match_score multidimensional:

`

$$\text{score} = w1\text{awareness} + w2\text{sof} + w3\text{jergas} + w4\text{deseo} + w5\text{avatar} + w6\text{temporalidad} + w7\text{fuerza_cambio} + w8\text{formato}$$

`

Pesos: awareness 0.20, sof 0.15, deseo 0.15, avatar 0.15, fuerza_cambio 0.10, jergas 0.10, temporalidad 0.10, formato 0.05.

3. Ordenar por score desc.

4. Seleccionar TOP 5 → tipo: similar.

5. Para el 6º (arriesgado), buscar el candidato con score 0.65-0.85 que tenga UN diferenciador claro (formato distinto, avatar lateral, ángulo nuevo) — el "wild card".

6. Si <6 candidatos válidos → FAIL y signal "relanzar buscadores con queries alternativas".

7. Generar por_que_similar y diferenciador (texto humano-legible) para cada uno.

8. Guardar _finalists.json.

QC crítico (heredado de agente-esceptico)

Antes de aprobar:

- ¿Los 5 similares son REALMENTE escalables o son chatarra "barata" en gethookd?
- ¿El arriesgado tiene un giro genuino o es solo "escogí el menos malo"?
- ¿Algún finalista es UGC básico problem→solution en mercado saturado? → RECHAZAR.

Si supervisor escéptico falla → revisar y/o relanzar buscadores.

Verifier L1

verify_finalists.py — exactamente 6 entries, 5 con tipo=similar, 1 con tipo=arriesgado, todos con match_score ≥ 0.6, todos con video_path existente.

Restricciones

- 5+1 OBLIGATORIO. Nunca 6 similares ni nunca 4 similares.
- Si <6 candidatos válidos → FAIL claro al supervisor.
- Modelo Opus 4.7 para scoring (en duda, modelo más alto).

CHECKLIST DE AUDITORÍA · Supervisor de Control de Calidad

El supervisor de este sub-proceso recorre estos checks por cada ejecución. Si CUALQUIERA falla → BREAK con razón concreta y el agente reintenta con el feedback inline (REGLA #66, max_retries=5).

1. El output existe en la sección `caja.comparador` y NO está vacío.
2. Verifier L1 determinista pasa: `verify_6_finalists.py` (5 similares + 1 arriesgado, todos `awareness-match`).
3. Supervisor L2 olfato (Haiku): Detectar si el comparador está siendo demasiado laxo o demasiado estricto.
4. REGLAs duras del agente `.md` NO violadas (ver sección 'Restricciones' del agente).
5. Si paralelo: el `shared_state` del cluster fue consultado antes de actuar y actualizado después.
6. Default Opus 4.7 obligatorio (msg 4658) — sin fallback Haiku para razonamiento.
7. Si BREAK → escalación L3 Opus + escribir caso al CBR del supervisor.

Cascada de escalación (Andon Tier · REGLA #146)

- L1 verifier determinista (Python · 0 tokens) — runs SIEMPRE.
- L2 supervisor Haiku olfato — runs SIEMPRE en paralelo, captura lo nuevo.
- L3 supervisor sub-proceso Opus — runs si L1 FAIL o L2 FLAG. Resuelve y añade al CBR del sub-proceso.
- L4 Master Mejora Continua Opus — runs si ≥ 3 L3 escaladas en 24h o cron diario. Cross-pattern detection.
- L5 Fer humano — runs solo si L4 propone REGLA nueva o caso sin precedente.

Referencia al Master PDF

Este sub-proceso opera dentro del flow definido en [_documentacion/PDF_MAESTRO_ESCALADO_VIDEO_v16.pdf](#) · Flow D "Concepto ganador - 5 similares + 1 fuera de caja". Cualquier conflicto entre este PDF y el master → manda el master.

Si Fer modifica el master (ej: añade flag "saturated", cambia "5+1" a "4+2", etc.), el cambio se propaga a este PDF en la próxima generación. Este PDF NO es source-of-truth — es destilado del `agent.md` + `factory_metadata.json` + master.